

# Detection and Isolation of Failures in Directed Networks of LTI Systems

Mohammad Amin Rahimian, Victor M. Preciado \*

**Abstract**—We propose a methodology to detect and isolate link failures in a weighted and directed network of identical multi-input multi-output LTI systems when only the output responses of a subset of nodes are available. Our method is based on the observation of jump discontinuities in the output derivatives, which can be explicitly related to the occurrence of link failures. The order of the derivative at which the jump is observed is given by  $r(d+1)$ , where  $r$  is the relative degree of each system's transfer matrix, and  $d$  denotes the distance from the location of the failure to the observation point. We then propose detection and isolation strategies based on this relation. Furthermore, we propose an efficient algorithm for sensor placement to detect and isolate any possible link failure using a small number of sensors. Available results from the theory of sub-modular set functions provide us with performance guarantees that bound the size of the chosen sensor set within a logarithmic factor of the smallest feasible set of sensors. These results are illustrated through elaborative examples and supplemented by computer experiments.

## I. INTRODUCTION

Fault Detection and Isolation (FDI) is an active area of research with a wide range of applications, such as power systems analysis [1], robotic networks [2], and security of cyber-physical systems [3]. In particular, reliability analysis of multi-agent networks is vital in many areas of engineering, since many critical infrastructures can be modeled as networked dynamical systems. In this context, the collective dynamics of a network of dynamic agents can be severely affected by network failures, resulting in undesirable behaviors. Hence, studying the effects of link and/or node failures on the network dynamics is of vital importance with a wide range of practical implications [4], [5]. On account of its relevance, there is a wide literature on (Failure Detection and Isolation) FDI techniques and engineering applications (see, for example, [6] and references therein).

The analysis and development of FDI techniques for networks of LTI systems is theoretically, as well as practically, appealing. Networks of LTI systems have been extensively investigated, since they encompass particularly relevant dynamics, such as linear agreement protocols [7], [8]. In this paper, we focus our attention to this class of systems and develop graph-theoretic tools for FDI. Our work is related to [9], in which Yoon and Tsumura use graph theory to draw interesting results about the stability margins of transfer

functions between pairs of input/output nodes in terms of the length of shortest paths. In [10], Rahimian et al. derive sufficient conditions for detectability and identifiability of link failures in terms of inter-nodal distances between link failures and observation points.

The main objective of this paper is to provide an explicit methodology for FDI in directed networks of LTI agents. Our algorithms are based on the analysis of discontinuities in the derivatives of the output responses of a subset of sensor nodes. Based on our results, we also provide an efficient sensor placement algorithm for FDI using a small number of sensors. Although finding the minimum number of sensors for FDI is a hard combinatorial problem, we provide an approximation algorithm with quality guarantees based on submodular set functions. In particular, we prove that our algorithm is a  $\log |\mathcal{E}| + 1$  approximation to the combinatorial FDI problem, where  $|\mathcal{E}|$  is the number of edges in the network.

The remainder of this paper is organized as follows. Section II begins with some preliminaries on graph and matrix theory, as well as the networked dynamics. In Section II-C, we introduce the model of dynamic network under consideration and formulate the detection and isolation problem. Section III contains the relationship between the location of link failures and discontinuities in the network output signals. In Subsections IV-A and IV-B, efficient algorithms are proposed to find an effective selection of observation nodes, for both detection and isolation problems. In Subsection IV-C, we use results from the theory of submodular set functions to derive performance guarantees for our algorithms. Illustrative examples and discussions in Section V elucidate the results followed by computer experiments on large random networks. Section VI concludes the paper. All proofs are included in the Appendix.

## II. PRELIMINARIES AND PROBLEM FORMULATION

Throughout the paper, the imaginary unit is denoted by  $J := \sqrt{-1}$ . The set of integers  $\{1, 2, \dots, k\}$  is denoted by  $[k]$ ,  $\mathbb{N}$  is the set of all positive integers,  $\mathbb{R}$  is the set of all real numbers, and any other set is represented by a calligraphic capital letter. The cardinality of a set  $\mathcal{X}$  is denoted by  $|\mathcal{X}|$ . The difference of two sets  $\mathcal{X}$  and  $\mathcal{Y}$  is denoted by  $\mathcal{X} \setminus \mathcal{Y}$ . Matrices are represented by capital letters and vectors are expressed by boldface lower-case letters. Moreover,  $\mathbf{1}_k$  and  $\mathbf{0}_k$  are  $k$ -dimensional column vectors with all ones and all zeros, respectively; and  $\mathbf{e}_{i,k}$  is the  $i$ -th vector in the standard basis of  $\mathbb{R}^k$ . For a matrix  $A$ ,  $a_{ij} := [A]_{ij}$  denotes its  $ij$ -th entry, and for a block matrix  $M$ ,  $M_{(i)(j)}$  denotes the matrix corresponding to the  $ij$ -th block. The  $k \times k$  identity matrix is denoted by  $I_k$ , and  $Z_k$  is the  $k \times k$  matrix of all zeros.

\* The authors are with the Department of Electrical and Systems Engineering, University of Pennsylvania, Philadelphia, PA 19104-6228 USA. (email: preciado@seas.upenn.edu).

This work was supported by the National Science Foundation grants CNS-1302222 “NeTS: Medium: Collaborative Research: Optimal Communication for Faster Sensor Network Coordination”, and IIS-1447470 “BIGDATA: Spectral Analysis and Control of Evolving Large Scale Networks”.

### A. Algebraic Graph Theory

A directed graph or *digraph*  $\mathcal{G}$  is defined as an ordered pair of sets  $\mathcal{G} := (\mathcal{V}, \mathcal{E})$ , where  $\mathcal{V} = \{\nu_1, \dots, \nu_N\}$  is a set of  $N = |\mathcal{V}|$  vertices (also called nodes or agents) and  $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$  is a set of directed edges (also called links or arcs). Each edge  $\epsilon := (\tau, \nu) \in \mathcal{E}$  is graphically represented by a directed arc from vertex  $\tau \in \mathcal{V}$  to vertex  $\nu \in \mathcal{V}$ . Vertices  $\nu$  and  $\tau$  are referred to as the *head* and *tail* of the edge  $\epsilon$ , and a  $(\nu, \nu)$  edge is called a self-loop on  $\nu$ . In our graphical representation, we do not allow parallel edges, also called multi-edges, and we also assume that graphs do not contain self-loops.

Given an integer  $k \in \mathbb{N}$ , an ordered set of (possibly repeated) indices  $(\alpha_1, \alpha_2, \dots, \alpha_k)$  with  $\alpha_i \in [N]$ , and two vertices  $\tau, \nu \in \mathcal{V}$ , a  $\tau\nu$ -walk of length  $k+1$  is defined as an ordered sequence of directed edges of the form  $\mathcal{W} := ((\tau, \nu_{\alpha_1}), (\nu_{\alpha_1}, \nu_{\alpha_2}), \dots, (\nu_{\alpha_{k-1}}, \nu_{\alpha_k}), (\nu_{\alpha_k}, \nu))$ . A cycle or closed walk on node  $\nu$  signifies a  $\nu\nu$ -walk. For any  $q, p \in [N]$ ,  $\Omega^k(\nu_q, \nu_p)$  is the set of all  $\nu_q\nu_p$ -walks in  $\mathcal{G}$  with length  $k$ . In the same venue, we define the distance from  $\nu_q$  to  $\nu_p$  in  $\mathcal{G}$  as

$$\text{dist}(\nu_q, \nu_p) = \min_{k \in \mathbb{N}, \Omega^k(\nu_q, \nu_p) \neq \emptyset} k,$$

where, by convention,  $\text{dist}(\nu_q, \nu_q) = 0$ ; and  $\text{dist}(\nu_q, \nu_p) = \infty$  if  $\Omega^k(\nu_q, \nu_p) = \emptyset$ , for all  $k \in \mathbb{N}$ . The diameter of  $\mathcal{G}$ , denoted by  $\text{diam}(\mathcal{G})$ , is defined as the maximum distance between any pair of nodes  $\nu_p, \nu_q \in \mathcal{V}$ :  $\text{diam}(\mathcal{G}) = \max_{\nu_p, \nu_q \in \mathcal{V}} \text{dist}(\nu_q, \nu_p)$ .

The adjacency matrix of  $\mathcal{G}$ , which we denote by  $G = [g_{ij}]$ , is a binary matrix  $G \in \{0, 1\}^{N \times N}$  such that  $G_{pq} = 0$  for all pairs  $p, q$  such that  $(\nu_q, \nu_p) \notin \mathcal{E}$  and  $G_{pq} = 1$ , otherwise. The following is a well-known result from algebraic graph theory [11], [12]:

**Lemma 1** (Weighted Walk Counting). *Given the adjacency matrix  $G$  of a directed graph, the following identities hold:*

- 1)  $[G^k]_{pq} = 0$ , for all  $k < \text{dist}(\nu_q, \nu_p)$ ,
- 2)  $[G^k]_{pq} > 0$  for  $k = \text{dist}(\nu_q, \nu_p)$ .

### B. Matrices and their Kronecker Algebra

For matrices  $A \in \mathbb{R}^{n \times m}$  and  $B \in \mathbb{R}^{p \times q}$ , their Kronecker product  $A \otimes B$  is defined as:

$$A \otimes B = \begin{bmatrix} a_{11}\mathbf{B} & \cdots & a_{1n}\mathbf{B} \\ \vdots & \ddots & \vdots \\ a_{m1}\mathbf{B} & \cdots & a_{mn}\mathbf{B} \end{bmatrix},$$

and given matrices  $M_1, M_2, M_3$  and  $M_4$  of appropriate dimensions, the following identities are always satisfied:

$$\begin{aligned} (M_1 \otimes M_2)(M_3 \otimes M_4) &= (M_1 M_3) \otimes (M_2 M_4), \\ (M_1 \otimes M_2) \otimes M_3 &= M_1 \otimes (M_2 \otimes M_3), \\ (M_1 \otimes M_2)^{-1} &= M_1^{-1} \otimes M_2^{-1}. \end{aligned} \quad (1)$$

In particular, for any  $k \in \mathbb{N}$ , (1) implies  $(M_1 \otimes M_2)^k = M_1^k \otimes M_2^k$ .

### C. Network Dynamic Model

Consider a network of  $N$  identical LTI subsystems whose interaction structure is expressed by a directed information

flow graph  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$  with adjacency matrix  $G = [g_{ij}]$ . Let  $\mathbf{x}_{(i)}(t) \in \mathbb{R}^d$  be the state of the system in node  $\nu_i$  at time  $t$ . The evolution of this subsystem for time  $t > t_0 \in \mathbb{R}$  is given by:

$$\dot{\mathbf{x}}_{(i)}(t) = A\mathbf{x}_{(i)}(t) + B \left( \sum_{q=1}^N g_{iq} \Gamma \mathbf{y}_{(q)}(t) + \mathbf{w}_{(i)}(t) \right), \quad (2)$$

$$\mathbf{y}_{(i)}(t) = C\mathbf{x}_{(i)}(t), \quad (3)$$

where  $\mathbf{y}_{(i)}(t) \in \mathbb{R}^o$  is the output of the  $i$ -th subsystem. Similarly,  $\mathbf{w}_{(i)}(t) \in \mathbb{R}^m$  is a vector of exogenous input signals injected into the  $i$ -th subsystem. Matrices  $A \in \mathbb{R}^{d \times d}$ ,  $B \in \mathbb{R}^{d \times m}$  and  $C \in \mathbb{R}^{o \times d}$  describe the evolution of each subsystem in isolation, and  $\Gamma \in \mathbb{R}^{m \times o}$  is called the inner-coupling matrix describing how the output of neighboring nodes influence the state of the  $i$ -th subsystem. We can describe the global network dynamics using a set of ‘stacked’ vectors,  $\mathbf{x} := (\mathbf{x}_{(1)}^T, \dots, \mathbf{x}_{(N)}^T)^T$ ,  $\mathbf{y} := (\mathbf{y}_{(1)}^T, \dots, \mathbf{y}_{(N)}^T)^T$  and  $\mathbf{w} := (\mathbf{w}_{(1)}^T, \dots, \mathbf{w}_{(N)}^T)^T$ , and Kronecker products, as follows,

$$\begin{aligned} \dot{\mathbf{x}}(t) &= (I_N \otimes A + G \otimes B \Gamma C) \mathbf{x}(t) + (I_N \otimes B) \mathbf{w}(t), \\ \mathbf{y}(t) &= (I_N \otimes C) \mathbf{x}(t). \end{aligned} \quad (4)$$

Part of our analysis will be performed in frequency domain. In this context, each individual subsystem can be represented as a  $o \times m$  transfer matrix  $H(s) = C(sI_d - A)^{-1}B$ . We denote by  $r$  the least relative degree among all the entries of  $H(s)$ , i.e., the least difference between the degrees of the polynomials in the denominator and the numerator of each entry. This quantity will be important in the succeeding derivations.

### D. Detection, Isolation, and Sensor Location Problems

Let us now state the particular problems considered in this paper. We describe the *detection problem* first. We assume that a central entity, which we call ‘designer’, has access to the outputs of a subset of nodes  $S \subseteq \mathcal{V}$ . We also assume that the designer knows the nominal network information flow digraph  $\mathcal{G}$  (the ‘faultless’ graph). Neither the location of the failure, nor the time of failure  $t_f$  are known by the designer. In the *failure detection problem*, the designer is interested in determining the existence of a single link failure, irrespective of its location, at any given time. In the *failure isolation problem*, however, the designer would like to determine, not only the existence of a failure, but also its location.

In this paper, we propose a detection and isolation algorithm based on the analysis of abrupt changes in the derivatives, up to a certain order  $z$ , of the sensor outputs induced by the failure, i.e., the designer has access to the derivatives,  $d^k \mathbf{y}_p / dt^k$  for  $k = 1, \dots, z$  and  $\nu_p \in S$ .

In the *sensor location problem*, the designer needs to choose the location of a set of sensors  $S \subseteq \mathcal{V}$  to be able to detect and isolate any potential edge failure for a given graph  $\mathcal{G}$ . The optimal solution of the problem is achieved when the designer solves the problem using the minimum number of sensors  $|S|$ . This problem is combinatorial in nature and can be shown to be NP-hard, by reductions to the set-cover problems [13]. In this paper, we propose a greedy algorithm to approximate

the optimal solution to this problem and provide quality guarantees using submodular set functions. In particular, we prove that our algorithm is a  $(\log |\mathcal{E}|) + 1$  approximation of the optimal combinatorial problem, in the sense that the cardinality of the sensor set returned by the proposed algorithm is no more than  $(\log |\mathcal{E}|) + 1$  times the minimum number of sensors required for the detection and isolation tasks.

### III. FAILURE DETECTION AND ISOLATION

In this Section we propose a methodology to detect and isolate link failures in the dynamic network model proposed in Subsection II-C. Our algorithm is based on the analysis of the derivatives of the output of sensor nodes  $S$  induced by a link failure. The impact on the dynamics of a particular link failure can be replicated by a carefully designed exogenous input, which we call the *fault-replicant input* and denote by  $\mathbf{f}(t)$ . In the rest of the section, we first analyze the effect of the fault-replicant input on the sensor measurements, in particular, on its derivatives (Subsection III-A). In Theorem 1, we provide an exact characterization of the effect of a faulty link on the sensor derivatives as a function of the distance, in number of hops, from the faulty link to the sensor node. Based on this characterization, in Subsection III-C we propose efficient algorithms to detect and isolate failures from the sensor outputs.

#### A. Modeling a Single-Link Failure

We start our analysis by considering the failure of a single link  $\bar{e} = (\nu_j, \nu_i) \in \mathcal{E}$  at time  $t_f > t_0$ . Consequently, the information flow graph for  $t > t_f$  is given by  $\bar{\mathcal{G}} = (\mathcal{V}, \mathcal{E} \setminus \{\bar{e}\})$ , which implies that  $g_{ij} = 0$  for  $t > t_f$ . Hence, the dynamics of the  $i$ -th subsystem (the ‘head’ of the faulty directed link  $\bar{e}$ ) for  $t > t_0$  is given by,

$$\dot{\mathbf{x}}_{(i)}(t) = A\mathbf{x}_{(i)}(t) + B \left( \sum_{q \neq j} g_{iq} \Gamma \mathbf{y}_{(q)}(t) + \mathbf{w}_{(i)}(t) \right). \quad (5)$$

It is convenient to replicate the dynamics of the network after link  $(\nu_j, \nu_i)$  fails by injecting an exogenous input  $\mathbf{f}_{(i)}(t)$  into the  $i$ -th node of the *faultless* network, as follows,

$$\begin{aligned} \dot{\mathbf{x}}_{(i)}(t) &= A\mathbf{x}_{(i)}(t) \\ &+ B \left( \sum_{q=1}^N g_{iq} \Gamma C \mathbf{x}_{(q)}(t) + \mathbf{f}_{(i)}(t) + \mathbf{w}_{(i)}(t) \right), \end{aligned}$$

where the exogenous, fault-replicant input is defined as

$$\mathbf{f}_{(i)}(t) := \begin{cases} 0, & \text{for } t < t_f, \\ -g_{ij} \Gamma C \mathbf{x}_{(j)}(t), & \text{for } t \geq t_f. \end{cases} \quad (6)$$

Notice that using  $\mathbf{f}_{(i)}(t) + \mathbf{w}_{(i)}(t)$  defined above as the exogenous input for the  $i$ -th subsystem results in the faulty dynamics in (5). We can incorporate the fault-replicant input into the faultless, global network dynamics in (4), by adding an exogenous input  $\mathbf{f}(t) := \mathbf{e}_{i,N} \otimes \mathbf{f}_{(i)}(t)$ . Since,  $\mathbf{x}_{(j)}(t) = (\mathbf{e}_{j,N}^T \otimes I_d) \mathbf{x}(t)$ , we have, from (6),

$$\begin{aligned} \mathbf{f}(t) &= \mathbf{e}_{i,N} \otimes (-g_{ij} \Gamma C (\mathbf{e}_{j,N}^T \otimes I_d) \mathbf{x}(t)) \\ &= -g_{ij} (\mathbf{e}_{i,N} \mathbf{e}_{j,N}^T \otimes \Gamma C) \mathbf{x}(t), \end{aligned}$$

for  $t \geq t_f$ ; and  $\mathbf{f}(t) = \mathbf{0}_{Nm}$  for  $t < t_f$ .

#### B. Impact of Single-Link Failures on Output Derivatives

We now proceed to analyze the effect of the fault-replicant input on the sensor measurements, in particular, on their derivatives. We present a theorem that characterizes the effect of single-link faults on the derivatives of the sensor outputs as a function of the distance from the faulty link to the sensor node. We state the theorem in terms of the following function,

$$\Delta_{p,k}(t) := \lim_{\varepsilon \rightarrow 0^+} \left( \left. \frac{d^k \mathbf{y}_{(p)}}{dt^k} \right|_{t+\varepsilon} - \left. \frac{d^k \mathbf{y}_{(p)}}{dt^k} \right|_{t-\varepsilon} \right),$$

which measures the jump in the  $k$ -th derivative of the output of node  $p$  at time  $t$ :  $\mathbf{y}_{(p)}(t)$ .

**Theorem 1** (Jump Discontinuities of Output Derivatives). *Consider the dynamic network in (2)-(3) where the exogenous input  $\mathbf{w}_{(i)}$  is  $(\text{diam}(\mathcal{G}) + 1)r$ -differentiable and  $r$  is the least relative degree among all the entries of the transfer matrix  $H(s) = C(sI_d - A)^{-1}B$ . Assume that link  $\bar{e} = (\nu_j, \nu_i)$  fails at time  $t_f$ . Then,  $\Delta_{p,k}(t_f) = \mathbf{0}_o$  for  $k < (\text{dist}(\nu_i, \nu_p) + 1)r$  and  $\Delta_{p,k}(t_f) = g_{ij} [G^k]_{pi} Q \Gamma C \mathbf{x}_{(j)}(t_f)$ , if  $k = (\text{dist}(\nu_i, \nu_p) + 1)r$ , where all distances are calculated w.r.t. the original digraph  $\mathcal{G}$ , and  $Q \in \mathbb{R}^{o \times o}$  is a constant matrix given by  $Q := \lim_{s \rightarrow \infty} s^k [H(s) \Gamma]^{\text{dist}(\nu_i, \nu_p) + 1}$ .*

**Remark 1.** *The above theorem characterizes the effect of a fault in link  $(\nu_j, \nu_i)$  on the  $k$ -th derivative of the output of node  $\nu_p$  as a function of the distance (in number of hops) from the head of the faulty link  $\nu_i$  to the output node  $\nu_p$ , and the relative degree  $r$  of the transfer matrix  $H(s)$ .*

#### C. Algorithms for Fault Detection and Isolation

In this subsection, we propose an algorithm for fault detection and isolation based on the preceding results. Theorem 1 states that the failure of link  $(\nu_j, \nu_i)$  induces a jump in the  $k$ -th derivative of the output of node  $\nu_p$  for  $k = (\text{dist}(\nu_i, \nu_p) + 1)r$ . We can therefore design a simple detection algorithm by monitoring the jumps in the derivatives of the sensor nodes. Specifically, if  $\Delta_{p,k}(t_f) \neq \mathbf{0}_o$  for any  $\nu_p \in S$ ,  $k \leq z$  and  $t_f > t_0$ , then a link has failed in the network at time  $t_f$ . The following condition for detectability of a link failure is a direct consequence of Theorem 1:

**Corollary 1** (Detectable Links). *The failure of link  $(\nu_j, \nu_i)$  is detectable using the set of output derivatives  $\{\frac{d^k}{dt^k} \mathbf{y}_{(p)} : \nu_p \in S, k \leq z\}$  if there exists a directed path of length  $l \leq (\frac{z}{r}) - 1$  from  $\nu_i$  to a node in  $S$ .*

Once a link failure is detected, it may also be isolated by looking at the values of  $p$  and  $k$  for which  $\Delta_{p,k}(t_f) \neq \mathbf{0}_o$ , under certain condition on the distribution of sensors, which we study in Section IV. Our strategy is based on exploiting the relationship between the location of the failure and abrupt changes in the output derivatives of a particular order. To achieve this, it is convenient to define a look-up table  $D$ , with  $d_{pe} = [D]_{pe} \in \{0, 1, \dots, z\}^{|S| \times |\mathcal{E}|}$ , with columns indexed by edges and rows by sensor nodes. Given a communication graph

$\mathcal{G}$ , the designer can compute the matrix  $D$ , running a double loop search over the set of edges  $\epsilon \in \mathcal{E}$  and the set of sensor nodes  $\nu_p \in S$ . For each edge  $\epsilon = (\nu_j, \nu_i)$  and node  $\nu_p$  in this loop, the designer assigns  $d_{p\epsilon} := (\text{dist}(\nu_p, \nu_i) + 1)r$ . Although matrix  $D$  is defined using distances in the communication graph, its entries has a dynamical interpretation. According to Theorem 1, if link  $\epsilon \in \mathcal{E}$  fails at time  $t_f$ , the we have

$$\min_{k \leq z} \{k : \Delta_{p,k}(t_f) \neq \mathbf{0}_o\} = d_{p\epsilon}.$$

In other words,  $d_{p\epsilon}$  is equal to the smallest order  $k$  of the derivative of sensor  $p$ 's output  $\mathbf{y}_p$ , at which we observe an abrupt change when link  $\epsilon$  fails. Therefore, using the matrix  $D$ , the designer may be able to isolate a link failure, as follows. First, when a link failure is detected at  $t = t_f$ , the designer constructs a column vector  $\mathbf{k} = (k_1, \dots, k_{|S|})^T$ , where  $k_p := \min_{k \leq z} \{k : \Delta_{p,k}(t_f) \neq \mathbf{0}_o\}$ , i.e.,  $k_p$  is the smallest order of the derivative at which the designer observes a jump in the output of sensor node  $\nu_p$ . Notice that, if the vector  $\mathbf{k}$  matches one, and only one, of the  $|\mathcal{E}|$  columns of  $D$ , then the designer can isolate the location of the link failure. In particular, if  $\mathbf{k}$  matches the  $\epsilon$ -th column of  $D$ , then the designer can conclude that link  $\epsilon$  has failed. In Section IV, we propose a sensor location algorithm to guarantee the existence and uniqueness of a column in  $D$  matching  $\mathbf{k}$  for any possible link failure. We illustrate the above methodology for failure isolation in the following example.

**Example 1. Isolation of a Link Failure.**

Consider the cycle network in Fig 1. Agents in this network are single integrators following a Laplacian dynamics, i.e.,  $\dot{\mathbf{x}} = -\mathcal{L}(\mathcal{G})\mathbf{x}$ , where  $\mathcal{L}(\mathcal{G})$  is the Laplacian matrix of the network. Assume that we have two sensor nodes,  $S = \{\nu_2, \nu_3\}$ , and the edges are labeled such that the edge set is given by  $\mathcal{E} = \{\epsilon_q, q \in [5]\}$ , where for  $q \in [5] \setminus \{1\}$ ,  $\epsilon_q = (\nu_{q-1}, \nu_q)$  and  $\epsilon_1 = (\nu_5, \nu_1)$ . Following the methodology proposed in this subsection, we can construct the following look-up table matrix,

$$D = \begin{pmatrix} 2 & 1 & 0 & 4 & 3 \\ 3 & 2 & 1 & 0 & 4 \end{pmatrix},$$

where the entries in the  $\epsilon$ -th column of  $D$ , for  $\epsilon = 1, \dots, 5$ , are the distances, plus one, from the head of the  $\epsilon$ -th link to each one of the nodes in  $S = \{\nu_2, \nu_3\}$ . Let us assume, for example, a failure in edge  $\epsilon_2 = (\nu_1, \nu_2)$ . According to Theorem 1, this failure induces abrupt changes in the 1<sup>st</sup> and 2<sup>nd</sup> derivatives of nodes 2 and 3, respectively, which results in a vector  $\mathbf{k} = (1, 2)^T$ . Notice that this vector matches the second column of  $D$ ; therefore, the designer is able to detect and isolate this failure from changes in the first two derivatives of the sensor nodes, by matching the order of derivatives in which the jumps are observed with the columns of  $D$ .

#### IV. EFFICIENT SENSOR PLACEMENT

It is not always possible to isolate, or even detect, link failures from a given set of sensor locations,  $S$ . In this section, we study the problem of distributing sensors in a network to guarantee that the designer is able to detect and isolate any possible link failure. Although finding the minimum number

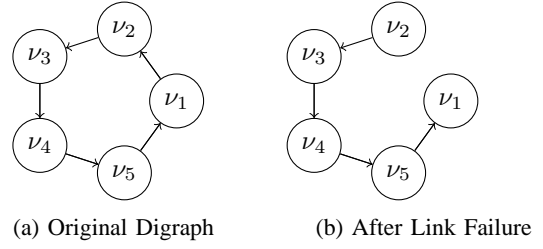


Fig. 1: A Directed Cycle of Length Five

of sensors to accomplish this goal is a hard combinatorial problem, we provide in this section efficient approximation algorithm with quality guarantees based on submodular set functions and the set-cover problem. In particular, we prove that our algorithm is a  $\log |\mathcal{E}| + 1$  approximation to the optimal combinatorial problem. In other words, the size of the sensor set  $S$  rendered by the approximation algorithm is at most  $\log |\mathcal{E}| + 1$  times the minimum number of sensors required for fault detection and isolation.

Before describing our sensor placement algorithm, we define a set of binary relations that will be used later on.

**Definition 1** (Binary Relations between Edges and Nodes). *We define the following binary relations between  $[N]$  and  $\mathcal{E}$ , denoted by  $\mathcal{R}_0$  and  $\mathcal{R}_k$ , for  $k \in [z]$ . For all  $p \in [N]$  and  $\epsilon = (\nu_q, \nu_s) \in \mathcal{E}$ :*

- $(p, \epsilon) \in \mathcal{R}_k \iff k = (\text{dist}(\nu_s, \nu_p) + 1)r$ ,
- $(p, \epsilon) \in \mathcal{R}_0 \iff (p, \epsilon) \notin \mathcal{R}_k \text{ for any } k \in [z]$ .

**Remark 2.**  $\mathcal{R}_k$  is defined such that if  $(p, \epsilon) \in \mathcal{R}_k$ , then the failure of link  $\epsilon$  produces a jump in the  $k$ -th derivative of the response of node  $p$ . On the other hand, if  $(p, \epsilon) \in \mathcal{R}_0$ , then the failure of edge  $\epsilon$  does not produce a jump in any of the derivatives of the response of node  $p$  up to the  $z$ -th order.

Based on the above definition, we propose in the following subsections efficient algorithms for sensor placement. We also provide quality guarantees based on the results developed in the context of set-cover problems [14], [15].

##### A. Detection of Link Failures: Coverage

The link-failure detection problem can be restated using the binary relation  $\mathcal{R}_0$ , as follows:

**Problem 1** (Detection). *Given a digraph  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ , find a subset of nodes  $\mathcal{M}_D \subseteq \mathcal{V}$  of minimum cardinality  $|\mathcal{M}_D|$ , such that for all  $\epsilon \in \mathcal{E}$ , there exists a node  $\nu_p \in \mathcal{M}_D$  with  $(p, \epsilon) \notin \mathcal{R}_0$ .*

Let  $\mathcal{M}_D$  be any solution to the above problem. Then, the failure of any link at time  $t_f$  would induce an abrupt change in some of the first  $z$  derivatives of the output of at least one node in  $\mathcal{M}_D$ . In other words, by observing the first  $z$  derivatives of the outputs of all nodes in  $\mathcal{M}_D$ , one is able to determine that a failure occurs. Finding  $\mathcal{M}_D$  is a hard combinatorial problem for which we propose an approximation algorithm below. To present this approximation, it is convenient to define the following concepts:

**Definition 2** (Submodular Functions). Given two sets  $\hat{\mathcal{M}}, \bar{\mathcal{M}} \subset \mathcal{V}$  such that  $\hat{\mathcal{M}} \subset \bar{\mathcal{M}}$ , a function<sup>1</sup>  $f : \mathcal{P}(\mathcal{V}) \rightarrow \mathbb{R}_+$  is submodular if for all  $\nu_q \in \mathcal{V}$  the following inequality holds true  $f(\hat{\mathcal{M}} \cup \{\nu_q\}) - f(\hat{\mathcal{M}}) \leq f(\bar{\mathcal{M}} \cup \{\nu_q\}) - f(\bar{\mathcal{M}})$ .

**Definition 3** (Coverage Function). Given a set of nodes  $\mathcal{M} \subset \mathcal{V}$ , we define the “coverage function”  $f_D : \mathcal{P}(\mathcal{V}) \rightarrow [|\mathcal{E}|] \cup \{0\}$ , as follows  $f_D(\mathcal{M}) = |\{\epsilon \in \mathcal{E} : \forall p \in \mathcal{M}, (p, \epsilon) \in \mathcal{R}_0\}|$ . In other words, this is the number of edges in the network whose failure would not induce a jump in any of the first  $z$  derivatives of the outputs of any node in  $\mathcal{M}$ .

**Claim 1** (Submodularity of Coverage Function). The function  $-f_D(\cdot)$  is a submodular set function from  $\mathcal{P}(\mathcal{V})$  to  $[|\mathcal{E}|] \cup \{0\}$ .

In what follows, we propose a greedy algorithm [16] to efficiently find an approximate solution  $\mathcal{M}_D$  to Problem 1. We provide performance guarantees in Subsection IV-C.

---

**Routine 1** Determine a Solution  $\mathcal{M}_D$  to Problem 1

---

**Input:**  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$

- 1:  $\mathcal{M}_D \leftarrow \emptyset$
- 2: **while**  $f_D(\mathcal{M}_D) \neq 0$  **do**
- 3:  $\nu_q \leftarrow \arg \min \{f_D(\mathcal{M}_D \cup \{\nu_q\}) - f_D(\mathcal{M}_D); \nu_q \in \mathcal{V} \setminus \mathcal{M}_D\}$
- 4:  $\mathcal{M}_D \leftarrow \mathcal{M}_D \cup \{\nu_q\}$
- 5: **end while**

**Output:**  $\mathcal{M}_D$

---

**Remark 3.** The function  $f_D(\mathcal{M}_D)$  measures the coverage of set  $\mathcal{M}_D$  by counting the number of links that are not yet covered by  $\mathcal{M}_D$ . At each iteration of Routine 1, the extra agent  $q$  is selected and added to  $\mathcal{M}_D$  such that the number of newly covered links is maximized. Note that since for any  $\epsilon = (\nu_q, \nu_p) \in \mathcal{E}$ ,  $(p, \epsilon) \in \mathcal{R}_1$  it follows that  $f_D(\mathcal{V}) = 0$ , whence Routine 1 is guaranteed to terminate.

The focus is next shifted to the isolation problem, for which a similar routine is developed.

### B. Isolation of Link Failures: Resolution

For analyzing the isolation problem, we introduce several definitions that will be useful to describe our isolation algorithm.

**Definition 4** (Indicator Set of an Edge). Given a subset of agents  $\mathcal{M} \subseteq \mathcal{V}$  and an edge  $\epsilon \in \mathcal{E}$ , we define the “indicator set” of  $\epsilon$  w.r.t.  $\mathcal{M}$  as the correspondence  $\mathcal{I} : \mathcal{P}(\mathcal{V}) \times \mathcal{E} \rightarrow \mathcal{P}([z] \cup \{0\} \times \mathcal{V})$ , given by  $\mathcal{I}(\mathcal{M}, \epsilon) = \{(k, \nu_p) \in ([z] \cup \{0\}) \times \mathcal{M} \text{ such that } (p, \epsilon) \in \mathcal{R}_k\}$ . In other words, given an edge  $\epsilon$  and a set of nodes  $\mathcal{M}$ , this indicator set is the set of all the pairs  $(k, \nu_p)$  where  $k$  is the order of the derivative at which there is a jump in the output of node  $\nu_p \in \mathcal{M}$  when edge  $\epsilon$  fails.

**Definition 5** (Resolution Function). Given a subset of agents  $\mathcal{M} \subseteq \mathcal{V}$ , define the set of unidentified edges associated

with  $\mathcal{M}$  as the correspondence  $\mathcal{U} : \mathcal{P}(\mathcal{V}) \rightarrow \mathcal{P}(\mathcal{E})$ , given by  $\mathcal{U}(\mathcal{M}) = \{\epsilon \in \mathcal{E} : \exists \hat{\epsilon} \in \mathcal{E} \setminus \{\epsilon\} \text{ for which } \mathcal{I}(\mathcal{M}, \epsilon) = \mathcal{I}(\mathcal{M}, \hat{\epsilon})\}$ . Defined such,  $\mathcal{U}(\mathcal{M})$  is the set of all edges whose failures are not uniquely identified based on the order of the derivative at which there is a jump in the output responses of the nodes in  $\mathcal{M}$ . We further define the “resolution function”,  $f_I(\mathcal{M}) = |\mathcal{U}(\mathcal{M})|$ , as the number of edges that are not uniquely identified based on the orders of the derivatives at which we observe jumps in the outputs of the nodes in  $\mathcal{M}$ .

**Claim 2** (Submodularity of Resolution Function). The function  $-f_I(\cdot)$  is a submodular set function from  $\mathcal{P}(\mathcal{V})$  to  $[|\mathcal{E}|] \cup \{0\}$ .

Next using the above concepts, we can restate the isolation problem, as follows:

**Problem 2** (Isolation). Given a digraph  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ , find a subset of vertices  $\mathcal{M}_I \subset \mathcal{V}$  with the smallest cardinality  $|\mathcal{M}_I|$ , such that  $f_I(\mathcal{M}_I) = 0$ .

**Remark 4.** Note that when  $f_I(\mathcal{M}_I) = 0$ , there are no edges in the network that cannot be uniquely identified based on the jumps in the derivatives of the outputs of nodes in  $\mathcal{M}_I$ . In other words, all edge failures can be isolated.

**Corollary 2.** Isolation of all edge failures is possible if and only if  $f_I(\mathcal{V}) = 0$ .

Since Problem 2 is hard to solve exactly, we now propose a greedy heuristic similar to the one in Subsection IV-A to find an approximate solution, as follows. We also provide quality guarantees in Subsection IV-C.

---

**Routine 2** Determine a Solution  $\mathcal{M}_I$  to Problem 2

---

**Input:**  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$  &  $\mathcal{M}_D$

- 1:  $\mathcal{M}_I \leftarrow \mathcal{M}_D$
- 2: **while**  $f_I(\mathcal{M}_I) \neq 0$  &  $\mathcal{M}_I \neq \mathcal{V}$  **do**
- 3:  $\nu_q \leftarrow \arg \min \{f_I(\mathcal{M}_I \cup \{\nu_q\}) - f_I(\mathcal{M}_I); \nu_q \in \mathcal{V} \setminus \mathcal{M}_I\}$
- 4:  $\mathcal{M}_I \leftarrow \mathcal{M}_I \cup \{\nu_q\}$
- 5: **end while**
- 6: **if**  $f_I(\mathcal{M}_I) \neq 0$  **then**
- 7:  $\mathcal{M}_I \leftarrow \emptyset$
- 8: **end if**

**Output:**  $\mathcal{M}_I$

---

**Remark 5.** Note that a solution  $\mathcal{M}_D$  to Problem 1 is required as an input for Routine 2, with which  $\mathcal{M}_I$  is initialized. This ensures that any valid output of Routine 2 also satisfies the detection requirements, and such an initialization is required because the  $\mathcal{R}_0$  relations (lack of jumps in some nodes) are just as informative for the isolation purposes. In other words, unlike the detection problem where the goal is to ensure a non-zero relation at one or more observation points for any edge in the network; in the case of isolation, a link may just as well be identified through its  $\mathcal{R}_0$  relations with all the observation points. This, however, renders the failure of the link in question undetectable and it is exactly to prevent such a case that the initialization step in Routine 2 is necessary.

<sup>1</sup>Given a set  $\mathcal{X}$ , we denote by  $\mathcal{P}(\mathcal{X}) = \{\mathcal{M} \mid \mathcal{M} \subset \mathcal{X}\}$  the power-set of  $\mathcal{X}$ , which is the set of all its subsets.

**Remark 6.** The function  $f_I(\mathcal{M}_I)$  measures the resolution of set  $\mathcal{M}_I$  by counting the number of links that are not uniquely identified through their relations  $\mathcal{R}_k$  with the vertexes of set  $\mathcal{M}_I$ . At each iteration of Routine 2, the extra agent  $q$  is selected and added to  $\mathcal{M}_I$  such that the resultant improvement in the resolution of  $\mathcal{M}_I$  is maximized. Note that unlike Problem 1, it is possible for Problem 2 to have no solutions at all, in which case Routine 2 returns  $\emptyset$ . This occurs if, and only if,  $f_I(\mathcal{V}) \neq 0$ .

### C. Performance Guarantees from the Set Covering Problem

The following definition is most useful when implementing Routines 1 and 2 in general networks and selecting possible solutions  $\mathcal{M}_D$  and  $\mathcal{M}_I$  for Problems 1 and 2, respectively. Label the edges of the network from 1 to  $l = |\mathcal{E}|$  and denote them by  $\mathcal{E} = \{\epsilon_q, q \in [l]\}$ . Associate with every edge  $\epsilon_q, q \in [l]$  a row vector  $\mathbf{r}_q$  with  $N$  columns, whose element  $[\mathbf{r}_q]_\gamma, \gamma \in [N]$  is equal to  $k$  if  $(\gamma, \epsilon_q) \in \mathcal{R}_k$  for some  $k \in [z] \cup \{0\}$  and let matrix  $R \in ([z] \cup \{0\})^{l \times N}$  be the matrix whose  $q$ -th row is equal to  $\mathbf{r}_q$  for all  $q \in [l]$ . The highest order of derivatives  $z$  that is to be observed at the observation points is then set equal to the maximum value of the entries of matrix  $R$  and is thus bounded above by the diameter of the network.

Problems 1 and 2 can now be restated in terms of the matrix  $R$  as follows. For the detection problem choose the columns of  $R$  such that the corresponding entries have at least one non-zero element per each row. By the same token, for the isolation choose the columns such that the set of entries ordered in the order of the chosen columns are not identical for any two rows.

Let the location of the non-zero entries of the matrix  $R$  of a network be indexed by the one entries in a binary matrix  $\tilde{R}$ , which has the same dimensions  $l \times N$  as  $R$ . Problem 1 can then be formulated as finding  $\mathbf{x}$  such that:

$$\begin{aligned} & \underset{\mathbf{x} \in \{0,1\}^N}{\text{minimize}} && \mathbb{1}_N^T \mathbf{x} \\ & \text{subject to} && \tilde{R}\mathbf{x} \geq \mathbb{1}_l^T. \end{aligned} \quad (7)$$

The discrete optimization problem in (7) is an instance of the set covering problem [17]. Accordingly, given equal column weights, it is desired to choose a subset of columns with minimum total weight, such that at least one non-zero element from each row is selected. For an integer  $d \in \mathbb{N}$ , define the truncated harmonic sum  $\mathcal{H}(d) = \sum_{i=1}^d \frac{1}{i}$  and let  $d_{max}$  be the maximum of the column sums associated with matrix  $\tilde{R}$ . It is known from the classical results [13], [18], [19], [20], that the greedy heuristic in Routine 1 is guaranteed to produce a result  $\mathcal{M}_D$ , which is no worse than  $\mathcal{H}(d_{max})$  times the value of the optimal answer  $\mathbb{1}_N^T \mathbf{x}$  obtained from 7.

Similarly for Problem 2, it lends itself to the following formulation as a sub-modular set covering problem [21],

$$\begin{aligned} & \underset{\mathcal{S} \subset [N]}{\text{minimize}} && \sum_{j \in \mathcal{S}} w_j \\ & \text{subject to} && f_I(\{\nu_q, q \in \mathcal{S}\}) = f_I(\mathcal{V}), \end{aligned}$$

where using the unity column weights:  $w_j = 1, \forall j \in [N]$ , the objective can be rewritten as  $\sum_{j \in \mathcal{S}} w_j = |\mathcal{S}|$ . It is again known from the classical theory that the value of the solution

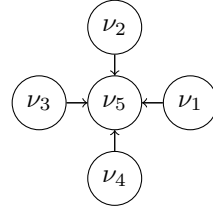


Fig. 2: A Star Digraph of Size Five

of the greedy heuristic in Routine 2 never exceeds the optimal value by more than a factor of  $\mathcal{H}(\max_j \{l - f_I(\{\nu_j\})\})$ , [21]. Noting the trivial bounds,  $\mathcal{H}(d_{max}) \leq H(|\mathcal{E}|)$ ,  $\mathcal{H}(\max_j \{l - f_I(\{\nu_j\})\}) \leq \mathcal{H}(|\mathcal{E}|)$ , and  $\mathcal{H}(|\mathcal{E}|) \leq \log(|\mathcal{E}|) + 1$ , it follows that the proposed sensor placement algorithms of Subsections IV-A and IV-B provide answers that are within a factor  $\log(|\mathcal{E}|) + 1$  of the optimal answer; and therefore, they scale with a rate no worse than the logarithm of the size of the network edge-set.

## V. EXAMPLES AND DISCUSSIONS

In this section we provide additional examples and discussions of special cases, followed by computer experiments with a random geometric graph.

### Example 2. Star Networks.

For this and the next subsection, let  $\mathcal{V} = \{\nu_q, q \in [5]\}$  be a set of five vertexes. As the first example, consider the case of a star network (Fig. 2), where there are four edges in the network and all of them share the same head vertex  $\nu_5$ . In this case, the designer can detect the failure of any single edge in the network by observing the first derivative of the response of  $x_5$ ; however, there are no subset of nodes that can be observed for isolating the failed edge. In fact  $f_I([5]) = 4$  in the case of a star network, and every edge of the network is in the same relations  $\mathcal{R}_1$  with the node  $\nu_5$  and  $\mathcal{R}_0$  with the rest of the nodes. Indeed, this is best seen through its  $R$  matrix given as  $R = (Z_4 \mathbb{1}_4)$ .

The above generalizes for finite star networks of arbitrary size. In particular, there is no solution to the isolation problem for any star network and detection can be achieved by observing the first derivative of the common head vertex.

### Example 3. Laplacian Dynamics on Cycle Networks.

As the second example, consider the five node cycle in Fig.1 and Example 1. The matrix  $R$  for this network is given by:

$$R = \begin{pmatrix} 1 & 2 & 3 & 4 & 0 \\ 0 & 1 & 2 & 3 & 4 \\ 4 & 0 & 1 & 2 & 3 \\ 3 & 4 & 0 & 1 & 2 \\ 2 & 3 & 4 & 0 & 1 \end{pmatrix}.$$

It is evident from matrix  $R$  that any two distinct vertexes offer a solution  $\mathcal{M}_I$  to Problem 1, since the locations of 0 entries do not overlap for distinct vertexes. Thence, the designer can detect the failure of any links in the cycle network by observing the jumps in the first four derivatives of any two nodes in the network. In the same vein, any set of two distinct

vertexes can also be used to uniquely determine which link has failed based on the observed jumps in the first four derivatives. For instance, taking  $\mathcal{M}_I = \{2, 3\}$ ,  $\epsilon_2$  is the only edge whose failure produces a jump in the first and second derivatives of  $x_2(t)$  and  $x_3(t)$ , respectively, at the time of failure,  $t = t_f$ . Fig 3 depicts the responses of the second and third agents, for a directed cycle of length five initialized at  $\mathbf{x}(0) = (1, 2, 3, 4, 5)^T$ , where for all  $t \in \mathbb{R}$ ,  $\mathbf{x}(t) = (x_1(t), x_2(t), x_3(t), x_4(t), x_5(t))^T$  and  $\dot{\mathbf{x}}(t) = -\mathcal{L}(\mathcal{G})\mathbf{x}(t)$ ,  $5 > t > 0$ , with  $\mathcal{L}(\mathcal{G})$  denoting the graph Laplacian, given by:

$$\mathcal{L}(\mathcal{G}) = \begin{pmatrix} 1 & 0 & 0 & 0 & -1 \\ -1 & 1 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 \\ 0 & 0 & 0 & -1 & 1 \end{pmatrix}.$$

The edge  $\epsilon_2$  is removed at time  $t_f = 5$ , whence the agents evolution becomes  $\dot{\mathbf{x}}(t) = -\mathcal{L}(\bar{\mathcal{G}})\mathbf{x}(t)$ ,  $t > 5$ , where  $\mathcal{L}(\bar{\mathcal{G}})$  is the same as  $\mathcal{L}(\mathcal{G})$  above except for its second row which is all zeros. Consequently, the first derivative of  $x_2(t)$  and the second derivative of  $x_3(t)$  exhibit jump discontinuities at  $t_f = 5$ , as depicted in Fig 4.

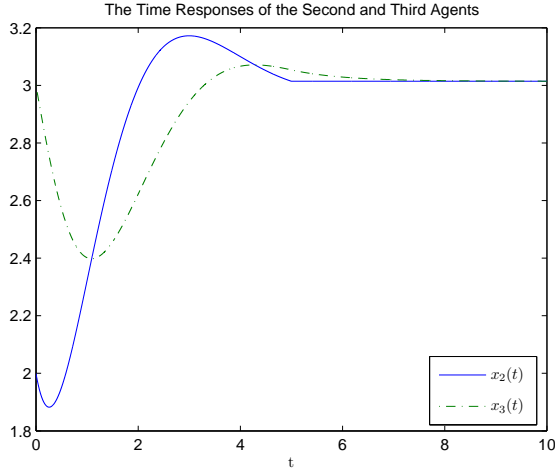


Fig. 3: The output responses  $x_2(t)$  and  $x_3(t)$  are plotted for  $0 \leq t \leq 10$ . The link failure happens at  $t_f = 5$ , where there is a break in the plot of  $x_2(t)$  but not of  $x_3(t)$ .

For any finite cycle network of arbitrary size, if all derivatives up to one less than the network size are observed, then any two nodes offer a solution to not only the detection, but also the isolation problem. The attention is next shifted to the sensor placement problem and the performance of the algorithms proposed in Subsections IV-A and IV-B.

**Example 4.** *Sensor Placement for Detection and Isolation in Simple Random Instances.*

Computer codes for Routines 1 and 2 are developed in MATLAB<sup>®</sup> and tested on randomly generated networks. Given the adjacency matrix of the network, a subroutine computes the matrix  $R$  defined at the beginning of this section, which is then used to compute the outputs of functions  $f_I$  and  $f_D$  for any given subset of nodes, as required for the

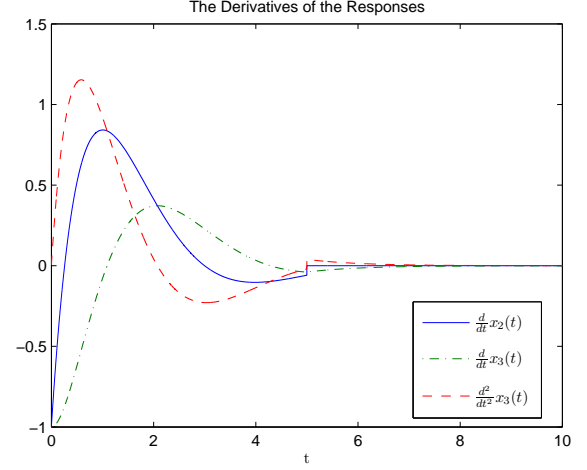


Fig. 4: The derivatives of  $x_2(t)$  and  $x_3(t)$  are plotted for  $0 \leq t \leq 10$ . At the time of failure  $t_f = 5$ , there are jump discontinuities in the plots of  $\frac{d}{dt}x_2(t)$  and  $\frac{d^2}{dt^2}x_3(t)$ . The plot of  $\frac{d^2}{dt^2}x_2(t)$  contains an impulse at  $t_f = 5$ , while  $\frac{d}{dt}x_3(t)$  is a continuous function of time.

implementation of the routines given in Subsections IV-A and IV-B. For an arbitrary graph input to Routines 1 and 2 several cases may arise. For the digraph in Fig. 5a, detection is achieved through the four nodes that are indicated with stars; however, the highlighted links cannot be isolated using the output of just these four nodes. In fact, increasing the set of observation points to include the entire vertex set still does not reduce the set of indistinguishable edges, and complete isolation of all links in this digraph is impossible.

The situation in the digraph of Fig. 5b is reversed, in the sense that with the same set of nodes that are the output of Routine 1 complete isolation is also achieved. In other words, no extra nodes are needed after  $\mathcal{M}_I$  is initialized with  $\mathcal{M}_D$  in Routine 2. For the digraph of Fig. 5c, on the other hand, with the detection output the two highlighted edges cannot be distinguished, but their status is resolved upon the addition of an extra node in Fig. 5d.

**Example 5.** *Computer Experiments with a Random Geometric Graph.*

In the following, the performance of the developed routines is tested for a random geometric graph model, where the nodes of the network are randomly and uniformly spread across a bounded region, and there is an edge wherever a certain distance threshold is met. The orientation of edges in each case is chosen by independent fair coin flips. The graph of Fig. 6 depicts one such graph instance with 50 nodes and 200 unidirectional edges. It follows that a total of 17 nodes is sufficient for detection, and these 17 nodes enable the isolation of all but 75 edges of the digraph, which are highlighted in Fig. 6. For this directed network, by observing all of the nodes in the network, the cardinality of the set of unresolved links reduces to 34.



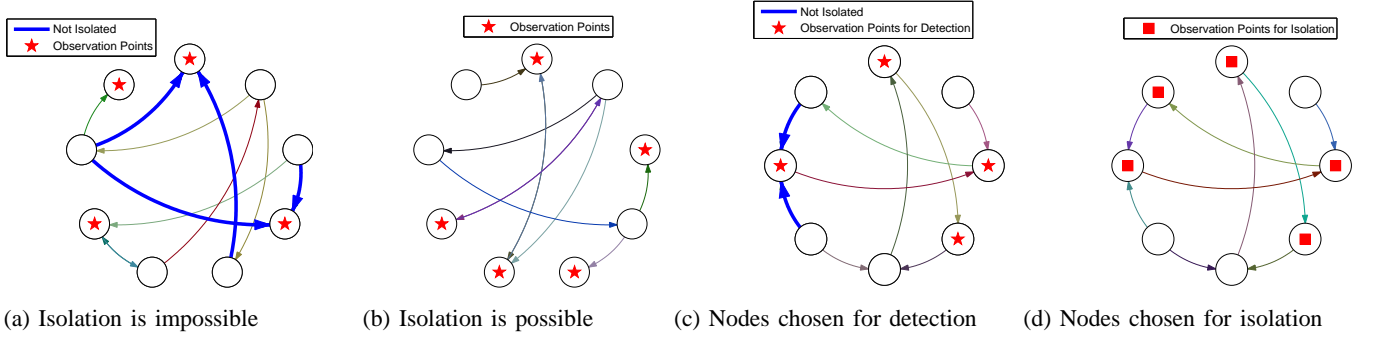


Fig. 5: (a) Even with all of the nodes selected as observation points, the highlighted links will remain unidentified. (b) The indicated nodes are enough to achieve both detection and isolation for all of the network links. (c) The highlighted links are not distinguishable using the indicated observation points. (d) By adding an extra node to the sensor set of (c), all edges become isolated.

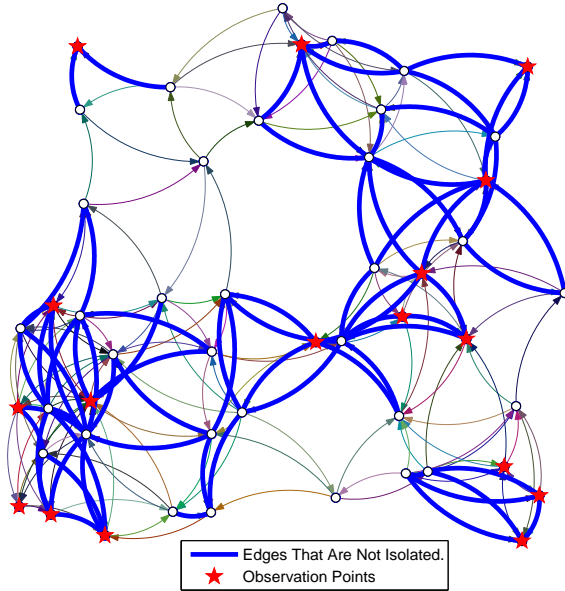


Fig. 6: By observing upto the 9-th orders of derivatives in the indicated 17 nodes, the 75 edges that are highlighted cannot be isolated, although every edge is detectable; and even with all of the nodes observed, there still remain 34 edges that cannot be isolated.

## VI. CONCLUSIONS

In this paper, a method was developed, both analytically and algorithmically, that enables the designer to detect and isolate link failures, based on the observed jumps in the derivatives of the output responses of a subset of nodes in a network of identical LTI systems. A theorem was presented, which relates the jumps in the derivatives at the time of failure to the distance of the failed link from the observation point. Next a set of efficient algorithms was developed for sensor placement, which together with the theorem, enables the designer to determine the existence and location of any link failure based on the observed jumps. Performance guarantees from the set coverage problem ensure that the proposed algorithms always

return a result that is with a  $\log |\mathcal{E}| + 1$  factor of the optimum answer, where  $E$  is the number of edges in the network.

The authors' ongoing research focuses on the development of detection techniques for jumps in high order derivatives using binary hypothesis testing, as well as the investigation of the case where the networked systems are heterogeneous and with bounds on the number of available sensors.

## APPENDIX A PROOFS FOR THE MAIN RESULTS

### A. Proof of Theorem 1: Jump Discontinuities of the Output Derivatives

Given the state space equations in (4), the transfer matrix from the input  $\mathbf{f}_{(i)}(t)$  to the output  $\mathbf{y}_{(p)}(t)$  can be written as:

$$\begin{aligned} [\mathbb{H}(s)]_{(p)(i)} &= (\mathbf{e}_{p,N}^T \otimes C) \times \dots \\ [sI_{Nd} - (I_N \otimes A + G \otimes B\Gamma C)]^{-1} (\mathbf{e}_{i,N} \otimes B) \\ &= (\mathbf{e}_{p,N}^T \otimes C) \times \dots \\ [(I_N \otimes (sI_d - A)) - (G \otimes B\Gamma C)]^{-1} (\mathbf{e}_{i,N} \otimes B) \end{aligned}$$

Pre- and post-multiplication by  $[I_N \otimes (sI_d - A)^{-1}]$  yields:

$$\begin{aligned} [\mathbb{H}(s)]_{(p)(i)} &= (\mathbf{e}_{p,N}^T \otimes C) [I_N \otimes (sI_d - A)^{-1}] \times \dots \\ \left\{ I_{Nd} - [G \otimes B\Gamma C (sI_d - A)^{-1}] \right\}^{-1} (\mathbf{e}_{i,N} \otimes B) \end{aligned}$$

The matrix inverse can be expanded as follows:

$$\begin{aligned} [\mathbb{H}(s)]_{(p)(i)} &= (\mathbf{e}_{p,N}^T \otimes C) [I_N \otimes (sI_d - A)^{-1}] \times \dots \\ \left\{ I_{Nd} + \sum_{l=1}^{\infty} [G \otimes B\Gamma C (sI_d - A)^{-1}]^l \right\} (\mathbf{e}_{i,N} \otimes B) \end{aligned}$$

Write

$$[G \otimes B\Gamma C (sI_d - A)^{-1}]^l = G^l \otimes (B\Gamma C (sI_d - A)^{-1})^l,$$



and apply Lemma 1 for the  $pi$ -th block of the center matrix, to get:

$$[\mathbb{H}(s)]_{(p)(i)} = \sum_{l=\text{dist}(\nu_i, \nu_p)}^{\infty} \{ [G^l]_{pi} C (sI_d - A)^{-1} \times \dots \\ (B\Gamma C (sI_d - A)^{-1})^l B \}.$$

Using  $H(s) = C(sI_d - A)^{-1}B$ , the above can be rewritten as:

$$[\mathbb{H}(s)]_{(p)(i)} \Gamma = \sum_{l=\text{dist}(\nu_i, \nu_p)}^{\infty} [G^l]_{pi} [H(s)\Gamma]^{l+1}.$$

From (3) the Laplace transform of the output at node  $p$  is given by:

$$\hat{\mathbf{y}}_{(p)}(s) = (\mathbf{e}_{p,N}^T \otimes I_o) \mathbb{H}(s) (\hat{\mathbf{f}}(s) + \mathbf{W}(s)) \\ + (\mathbf{e}_{p,N}^T \otimes C) [sI_{Nd} - (I_N \otimes A + G \otimes B\Gamma C)]^{-1} \mathbf{x}(t_0) e^{-t_0 s} \quad (8)$$

Note that Laplace transforms of outputs in the the healthy and faulty systems differ only in the term  $\hat{\mathbf{f}}(s)$  appearing in (8). To measure the jump in the  $k$ -th derivative of  $\mathbf{y}_{(p)}(t)$  due to the link failure stimulated by the virtual input  $\mathbf{f}(t)$ , we apply the initial value theorem to the  $k$ -th derivatives of the time shifted outputs for the healthy and faulty systems and calculate their difference to get:

$$\Delta(p, k) = \lim_{s \rightarrow \infty} s^{k+1} e^{t_f s} (\mathbf{e}_{p,N}^T \otimes I_o) \mathbb{H}(s) \hat{\mathbf{f}}(s) \\ = \lim_{s \rightarrow \infty} s^{k+1} e^{t_f s} [\mathbb{H}(s)]_{(p)(i)} \hat{\mathbf{f}}_{(i)}(s),$$

which can be decomposed as

$$\Delta(p, k) = \\ \left( \lim_{s \rightarrow \infty} s^k [\mathbb{H}(s)]_{(p)(i)} \right) \left( \lim_{s \rightarrow \infty} s e^{t_f s} \hat{\mathbf{f}}_{(i)}(s) \right). \quad (9)$$

The remainder of the proof is in calculating the two limits appearing in (9). We use Laplace domain techniques along with facts from complex analysis to calculate  $\lim_{s \rightarrow \infty} s e^{t_f s} \hat{\mathbf{f}}_{(i)}(s)$ , which we shall see is contributing a constant multiplier factor to the quantity  $\Delta(p, k)$ . On the other hand, the calculation of  $\lim_{s \rightarrow \infty} s^k [\mathbb{H}(s)]_{(p)(i)}$  is based primarily on Lemma 1 and that is where the inter-nodal distance relations come into play.

We begin by taking the Laplace transform of  $\mathbf{f}_{(i)}(t)$  in (6):

$$\hat{\mathbf{f}}_{(i)}(s) = -g_{ij} \Gamma C \int_0^{+\infty} \mathbf{x}_{(j)}(t) \bar{H}_{t_f}(t) e^{-st} dt, \quad (10)$$

where  $\bar{H}_{t_f}(t)$  is the shifted Heaviside step function given by  $\bar{H}_{t_f}(t) = 0$  for  $t < t_f$  and  $\bar{H}_{t_f}(t) = 1$  otherwise. The Laplace transform of  $\bar{H}_{t_f}(\cdot)$  is given by  $\hat{H}_{t_f}(s) = e^{-t_f s}/s$  for  $s \in \mathbb{C}$  having positive real part. Using the contour integral for inverse Laplace transform [22, Chapter 7], we can write

$$\bar{H}_{t_f}(t) = \frac{1}{2\pi j} \int_{\sigma-j\infty}^{\sigma+j\infty} \frac{e^{-t_f \zeta}}{\zeta} d\zeta, \quad (11)$$

for  $\sigma$  large enough such that the vertical line from  $\sigma - j\infty$  to  $\sigma + j\infty$  lies to the right of all singularities of the integrand  $e^{-t_f \zeta}/\zeta$ . Next replacing (11) in  $\mathbf{L}(s) := \int_0^{+\infty} \mathbf{x}(t) \bar{H}_{t_f}(t) e^{-st} dt$  and switching the order of the inte-

grals yields

$$\mathbf{L}(s) = \frac{1}{2\pi j} \int_{\sigma-j\infty}^{\sigma+j\infty} \frac{e^{-t_f \zeta}}{\zeta} \int_0^{+\infty} \mathbf{x}_{(j)}(t) e^{-(s-\zeta)t} dt d\zeta. \quad (12)$$

Using the Laplace transform

$$\hat{\mathbf{x}}_{(j)}(s - \zeta) = \int_0^{+\infty} \mathbf{x}_{(j)}(t) e^{-(s-\zeta)t} dt,$$

and the change of variable  $\eta = s - \zeta$ , (12) can be rewritten as

$$\mathbf{L}(s) = \frac{1}{2\pi j} \int_{\sigma'-j\infty}^{\sigma'+j\infty} \frac{e^{-t_f(s-\eta)}}{\eta - s} \hat{\mathbf{x}}_{(j)}(\eta) d\eta \\ = \frac{e^{-t_f s}}{2\pi j} \int_{\sigma'-j\infty}^{\sigma'+j\infty} \frac{e^{t_f \eta}}{\eta - s} \hat{\mathbf{x}}_{(j)}(\eta) d\eta,$$

for  $\sigma'$  large enough. Wherefore we get that

$$\lim_{s \rightarrow \infty} s e^{t_f s} \mathbf{L}(s) = \frac{1}{2\pi j} \int_{\sigma'-j\infty}^{\sigma'+j\infty} \lim_{s \rightarrow \infty} \frac{s e^{t_f \eta}}{\eta - s} \hat{\mathbf{x}}_{(j)}(\eta) d\eta \\ = \frac{-1}{2\pi j} \int_{\sigma'-j\infty}^{\sigma'+j\infty} e^{t_f \eta} \hat{\mathbf{x}}_{(j)}(\eta) d\eta = -\mathbf{x}_{(j)}(t_f), \quad (13)$$

exchanging the limit and integral in the first equality justifiable through the dominated convergence, and the last equality following by the inverse Laplace transform. Using (13) in (10) thus establishes that

$$\lim_{s \rightarrow \infty} s e^{t_f s} \hat{\mathbf{f}}_{(i)}(s) = g_{ij} \Gamma C \mathbf{x}_{(j)}(t_f). \quad (14)$$

Next for the limit term,  $\lim_{s \rightarrow \infty} s^k [\mathbb{H}(s)]_{(p)(i)}$ , appearing in (9) we have

$$\lim_{s \rightarrow \infty} s^k [\mathbb{H}(s)]_{(p)(i)} \\ = \lim_{s \rightarrow \infty} s^k g_{ij} \left[ \sum_{l=\text{dist}(\nu_i, \nu_p)}^{\infty} [G^l]_{pi} [H(s)\Gamma]^{l+1} \right] \\ = g_{ij} \left[ \sum_{l=\text{dist}(\nu_i, \nu_p)}^{\infty} [G^l]_{pi} \lim_{s \rightarrow \infty} s^k [H(s)\Gamma]^{l+1} \right]. \quad (15)$$

However, with  $k = r(\text{dist}(\nu_i, \nu_p) + 1)$  in (15), it follows by the choice of  $r$  being the relative degree that  $\lim_{s \rightarrow \infty} s^k [H(s)\Gamma]^{l+1} := Q \in R^{o \times o}$  for  $l = \text{dist}(\nu_i, \nu_p)$  and  $\lim_{s \rightarrow \infty} s^k [H(s)\Gamma]^{l+1} = Z_o$  for  $l > \text{dist}(\nu_i, \nu_p)$ ; while for  $k < r(\text{dist}(\nu_i, \nu_p) + 1)$ ,  $\lim_{s \rightarrow \infty} s^k [H(s)\Gamma]^{l+1} = Z_o$  for all  $l \geq \text{dist}(\nu_i, \nu_p)$ . In particular, we have that

$$\lim_{s \rightarrow \infty} s^k [\mathbb{H}(s)]_{(p)(i)} \\ = \begin{cases} [G^k]_{pi} Q, & \text{if } k = r(\text{dist}(\nu_i, \nu_p) + 1), \\ Z_o, & \text{if } k < r(\text{dist}(\nu_i, \nu_p) + 1), \end{cases} \quad (16)$$

and the claim follows upon replacing (14) and (16) in (9). ■

### B. Claim 1: Submodularity of Coverage Function

For the proof we need the additional concept of a coverage function, and we use a known result from the theory of submodular set functions that coverage functions are submodular [23]. Given a collection of subsets of edges  $E_1, E_2, \dots, E_N \subset \mathcal{E}$ , where each  $E_q$  is associated with a node  $\nu_q$ , the coverage function  $f_C : \mathcal{P}(\mathcal{V}) \rightarrow \mathbb{R}_+$  is defined as  $f_C(S) = |\cup_{\nu_q \in S} E_q|$  for any  $S \subseteq \mathcal{V}$ . Given a node  $\nu_q$ , let  $C(\nu_q) = \{\epsilon \in \mathcal{E} : (q, \epsilon) \notin \mathcal{R}_0\}$  be the correspondence that for each node  $\nu_q$  in the network gives the set of edges whose failure does induce a jump in any of the first  $z$  derivatives of  $\mathbf{y}_q(t)$ . We have that  $f_D(\mathcal{M}) = |\cap_{\nu_q \in \mathcal{M}} C_q^c| = |(\cup_{\nu_q \in \mathcal{M}} C_q)^c| = |\mathcal{E}| - |\cup_{\nu_q \in \mathcal{M}} C_q|$ , where  $^c$  denotes the set complement w.r.t.  $\mathcal{E}$ . The claim now follows upon noting that the latter term is a coverage function. ■

### C. Claim 2: Submodularity of Resolution Function

Consider any two subsets of nodes  $\bar{\mathcal{M}}$  and  $\hat{\mathcal{M}}$  such that  $\bar{\mathcal{M}} \subset \hat{\mathcal{M}} \subset \mathcal{V}$  and a vertex  $\nu_q \in \mathcal{V} \setminus (\bar{\mathcal{M}} \cup \hat{\mathcal{M}})$ . Note that  $\bar{\mathcal{M}} \subset \hat{\mathcal{M}}$  implies that  $\mathcal{U}(\hat{\mathcal{M}}) \subset \mathcal{U}(\bar{\mathcal{M}})$  so that  $f_I(\hat{\mathcal{M}}) \leq f_I(\bar{\mathcal{M}})$ . In particular,  $f_I(\bar{\mathcal{M}}) \geq f_I(\bar{\mathcal{M}} \cup \{\nu_q\})$ . Now, take any  $\epsilon \in \mathcal{U}(\hat{\mathcal{M}})$  and note that there exists an  $\hat{\epsilon} \in \mathcal{U}(\hat{\mathcal{M}})$  such that  $\mathcal{I}(\bar{\mathcal{M}}, \epsilon) = \mathcal{I}(\bar{\mathcal{M}}, \hat{\epsilon})$ . Now if all such  $\epsilon$  and  $\hat{\epsilon}$  are in the same relations  $\mathcal{R}_k, k \in [z] \cup \{0\}$  with the vertex  $\nu_q$ , then  $f_I(\hat{\mathcal{M}} \cup \{\nu_q\}) - f_I(\hat{\mathcal{M}}) = 0 \geq f_I(\bar{\mathcal{M}} \cup \{\nu_q\}) - f_I(\bar{\mathcal{M}})$ . Next consider the case where  $\mathcal{U}(\hat{\mathcal{M}}) \setminus \mathcal{U}(\bar{\mathcal{M}} \cup \{\nu_q\}) \neq \emptyset$ . Any edge  $\epsilon \in \mathcal{U}(\hat{\mathcal{M}}) \setminus \mathcal{U}(\bar{\mathcal{M}} \cup \{\nu_q\})$  is such that for all edges  $\hat{\epsilon} \in \mathcal{U}(\hat{\mathcal{M}})$  having  $\mathcal{I}(\bar{\mathcal{M}}, \epsilon) = \mathcal{I}(\bar{\mathcal{M}}, \hat{\epsilon})$ ,  $\epsilon$  and  $\hat{\epsilon}$  satisfy different binary relations with the added vertex  $\nu_q$ . But then since both  $\epsilon, \hat{\epsilon} \in \mathcal{U}(\bar{\mathcal{M}})$  and are in different relations with  $\nu_q$  it follows that  $\epsilon, \hat{\epsilon} \in \mathcal{U}(\bar{\mathcal{M}}) \setminus \mathcal{U}(\bar{\mathcal{M}} \cup \{\nu_q\})$ . Hence, we have shown that  $\mathcal{U}(\hat{\mathcal{M}}) \setminus \mathcal{U}(\bar{\mathcal{M}} \cup \{\nu_q\}) \subset \mathcal{U}(\bar{\mathcal{M}}) \setminus \mathcal{U}(\bar{\mathcal{M}} \cup \{\nu_q\})$  so that again  $|\mathcal{U}(\hat{\mathcal{M}}) \setminus \mathcal{U}(\bar{\mathcal{M}} \cup \{\nu_q\})| = f_I(\hat{\mathcal{M}}) - f_I(\bar{\mathcal{M}} \cup \{\nu_q\}) \leq f_I(\bar{\mathcal{M}}) - f_I(\bar{\mathcal{M}} \cup \{\nu_q\}) = |\mathcal{U}(\bar{\mathcal{M}}) \setminus \mathcal{U}(\bar{\mathcal{M}} \cup \{\nu_q\})|$ , and the property of submodularity for  $-f_I(\cdot)$  follows by definition. ■

### REFERENCES

- [1] W. Pan, Y. Yuan, H. Sandberg, J. Gonçalves, and G.-B. Stan, "Real-time fault diagnosis for large-scale nonlinear power networks," in *Proceedings of the 52nd IEEE Conference on Decision and Control*, 2013, pp. 2340–2345.
- [2] M. Mesbahi and M. Egerstedt, *Graph Theoretic Methods in Multiagent Networks*. Princeton University Press, 2010.
- [3] F. Pasqualetti, F. Dorfler, and F. Bullo, "Attack detection and identification in cyber-physical systems," *Automatic Control, IEEE Transactions on*, vol. 58, no. 11, pp. 2715–2729, 2013.
- [4] M. A. Rahimian and A. G. Aghdam, "Structural controllability of multi-agent networks: Robustness against simultaneous failures," *Automatica*, vol. 49, no. 11, pp. 3149–3157, 2013.
- [5] J. Kleinberg, M. Sandler, and A. Slivkins, "Network failure detection and graph connectivity," *SIAM Journal on Computing*, vol. 38, no. 4, pp. 1330–1346, 2008.
- [6] J. Chen and R. J. Patton, *Robust Model-Based Fault Diagnosis for Dynamic Systems*. Springer Publishing Company, Incorporated, 2012.
- [7] R. Olfati-Saber and R. Murray, "Consensus problems in networks of agents with switching topology and time-delays," *IEEE Transactions on Automatic Control*, vol. 49, no. 9, pp. 1520–1533, 2004.
- [8] W. Ren and R. Beard, *Distributed Consensus in Multi-vehicle Cooperative Control*. Springer, 2008.
- [9] M.-G. Yoon and K. Tsumura, "Transfer function representation of cyclic consensus systems," *Automatica*, vol. 47, no. 9, pp. 1974–1982, Sep. 2011.
- [10] M. A. Rahimian, A. Ajorlou, and A. G. Aghdam, "Digraphs with distinguishable dynamics under the multi-agent agreement protocol," *Asian Journal of Control*, 2014, in press.
- [11] N. Biggs, *Algebraic Graph Theory*. Cambridge University Press, 1994.
- [12] V. M. Preciado and A. Jadbabaie, "Moment-based spectral analysis of large-scale networks using local structural information," *IEEE/ACM Transactions on Networking*, vol. 21, no. 2, pp. 373–382, Apr. 2013.
- [13] V. Chvatal, "A greedy heuristic for the set-covering problem," *Mathematics of Operations Research*, vol. 4, no. 3, pp. 233–235, 1979.
- [14] S. S. Dhillon and K. Chakrabarty, "Sensor placement for effective coverage and surveillance in distributed sensor networks," in *Proceedings of IEEE Wireless Communications and Networking Conference*, 2003, pp. 1609–1614.
- [15] J. Leskovec, A. Krause, C. Guestrin, C. Faloutsos, J. VanBriesen, and N. Glance, "Cost-effective outbreak detection in networks," in *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge discovery and Data Mining*, 2007, pp. 420–429.
- [16] A. Krause and C. Guestrin, "Near-optimal observation selection using submodular functions," in *Proceedings of the 22nd National Conference on Artificial Intelligence - Volume 2*. AAAI Press, 2007, pp. 1650–1654.
- [17] D. Bertsimas and R. Vohra, "Rounding algorithms for covering problems," *Mathematical Programming*, vol. 80, no. 1, pp. 63–89, 1998.
- [18] D. S. Johnson, "Approximation algorithms for combinatorial problems," *Journal of Computer and System Sciences*, vol. 9, no. 3, pp. 256–278, 1974.
- [19] L. Lovász, "On the ratio of optimal integral and fractional covers," *Discrete Mathematics*, vol. 13, no. 4, pp. 383–390, 1975.
- [20] G. Dobson, "Worst-case analysis of greedy heuristics for integer programming with nonnegative data," *Mathematics of Operations Research*, vol. 7, no. 4, pp. 515–531, 1982.
- [21] L. Wolsey, "An analysis of the greedy algorithm for the submodular set covering problem," *Combinatorica*, vol. 2, no. 4, pp. 385–393, 1982.
- [22] J. Brown and R. Churchill, *Complex variables and applications*. McGraw-Hill Higher Education, 2009.
- [23] A. Schrijver, *Combinatorial Optimization*. Springer, 2003.



**Mohammad Amin Rahimian** is a recipient of gold medal in 2004 Iran National Chemistry Olympiad. He was awarded an honorary admission to Sharif University of Technology, where he received his B.Sc. in Electrical Engineering-Control. In 2012, he received his M.A.Sc. in Electrical and Computer Engineering at Concordia University, and he is currently a PhD student at the GRASP Laboratory, University of Pennsylvania. His research interests include systems dynamics and network theory, control of multi-agent networks, localization of mobile sensors, and tuning of fractional controllers.



**Victor M. Preciado** received the Ph.D. degree in electrical engineering and computer science from the Massachusetts Institute of Technology, Cambridge, in 2008.

He is currently the Raj and Neera Singh Assistant Professor of Electrical and Systems Engineering at the University of Pennsylvania. He is a member of the Networked and Social Systems Engineering (NETS) program and the Warren Center for Network and Data Sciences. His research interests include network science, dynamic systems, control theory, complexity, and convex optimization with applications in social networks, technological infrastructure, and biological systems.

Random Geometric Graph with Random Edge Orientations

